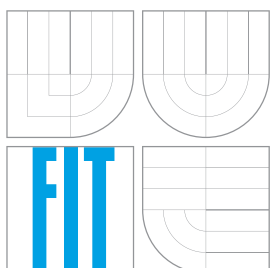


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZŠÍŘENÁ REALITA PRO VÝUKU HRY NA KLAVÍR

AUGMENTED REALITY FOR KEYBOARD PLAYING ASSISTANCE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN SUCHÁNEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MAREK ŠOLONY

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Suchánek Jan**

Obor: Informační technologie

Téma: **Rozšířená realita pro výuku hry na klavír/klávesy
Augmented Reality for Keyboard Playing Assistance**

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s AR knihovnamy, problematikou homografie a s principy rozšířené reality.
2. Navrhněte systém, který pomocí kamery a projektoru bude zobrazovat akordy/stupnice přímo na hudebním nástroji.
3. Daný systém implementujte a testujte.
4. Výsledky shrňte, vytvořte plakát demonstrující daný systém a uveďte možnosti rozšíření a využití v praxi.

Literatura:

- Bradski G. R., Kaehler A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc. 2008.
- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Šolony Marek, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se věnuje tématu rozšířené reality a jejímu využití pro výuku hry na klavír nebo klávesy. Je popsán návrh a implementace prototypu výukového systému, složeného z projektoru, počítače a kamery, který interaktivně podporuje žáka ve hře a umožňuje hrát na klavír bez předchozí znalosti not. K tomu systém využívá barevného zvýrazňování určitých kláves nástroje. Prototyp využívá knihovnu OpenCV a framework Qt.

Abstract

This bachelor thesis is focused topic of augmented reality and its practical usage for keyboard playing assistance. In this document is described design and implementation of application prototype. This prototype uses projector and camera to highlight specific keys of instrument and allow player to play without musical knowledges.

Klíčová slova

OpenCV, Rozšířená realita, MIDI, Homografie.

Keywords

OpenCV, Augmented reality, MIDI, Homography.

Citace

SUCHÁNEK, Jan. *Rozšířená realita pro výuku hry na klavír*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Šolony Marek.

Rozšířená realita pro výuku hry na klavír

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana inženýra Marka Šolonyho. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Suchánek
17. května 2016

Poděkování

Tímto bych chtěl poděkovat všem, kteří mě při psaní práce podporovali. V první řadě vedoucímu práce panu Ing. Marku Šolonymu za odborné konzultace a obětavost. Pak panu Vojtěchu Klučkovi za pomoc při uchycení projektoru a výrobu úchytu.

© Jan Suchánek, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
1.1	Motivace – Rozšířená realita	3
1.2	Existující řešení	4
1.3	Porovnání hardware	6
1.3.1	Zobrazení rozšířené reality	6
1.3.2	Příjem zvuku	6
2	Teorie	8
2.1	Projektivní geometrie	8
2.1.1	Transformace	8
2.1.2	Homogenní souřadnice	9
2.2	Homografie	9
2.2.1	Mapování pomocí homografie	9
2.3	Počítačové vidění	11
2.4	MIDI	11
2.4.1	Komunikace s PC	11
3	Návrh řešení	12
3.1	Popis systému	12
3.2	Zvýraznění kláves	13
3.3	Zpětná vazba od nástroje	14
4	Implementace	15
4.1	Model kláves	15
4.2	Zpětná vazba	16
4.3	Vnitřní reprezentace skladeb	17
4.4	Pomocné třídy	17
4.5	Třídy uživatelského rozhraní	18
4.5.1	Třída <code>MainWindow</code>	18
4.5.2	Třída <code>CameraWindow</code>	19
4.5.3	Třída <code>ProjectorWindow</code>	20
4.5.4	Třída <code>DragLabel</code>	20
4.6	Technologie	21
5	Testování	22
5.1	Uživatelské testování	22
5.1.1	Herní úkoly	22
5.2	Uživatelské hodnocení	23

5.3 Odhalené nedostatky	24
5.4 Další vylepšení	24
6 Závěr	26
Literatura	27
Přílohy	28
Seznam příloh	29
A Obsah CD	30

Kapitola 1

Úvod

V dnešní době se často objevují slova, jako virtuální realita a rozšířená realita. Oba tyto termíny znějí moderně a jsou populární hlavně mezi mladšími, pro které je kombinace herního a reálného světa nejvíce lákavá. Toho můžeme i využít. Pomocí interaktivní výuky je snazší žáky zaujmout a ti pak mají větší zájem o danou činnost.

Cílem práce je navrhnout systém, který využívá rozšířenou realitu pro podporu výuky a umožňuje začít s hrou na klavír i hráčům bez znalostí hudební teorie. Systém pomocí projektoru a kamery bude schopen názorně ukázat, které klávesy je třeba zahrát. Protože jde o výukový systém, mělo by být umožněno učenou skladbu přehrát po částech, případně různě manipulovat s rychlostí přehrávání, aby měl hráč šanci naučit se a zapamatovat si sled hraných kláves.

Tato práce je členěna do následujících kapitol. V kapitole 2 je popsán teoretický základ práce. Je vysvětlena projektivní geometrie a její použití v počítačovém vidění. Je také přiblížen pojem homografie. Věnuje se i standardu MIDI. V kapitole 3 je popsán návrh systému. Jsou popsány jednotlivé části hardwaru, které byly použity pro sestavení systému a je vysvětlen princip, jak celý systém funguje. Kapitola 4 popisuje samotnou implementaci aplikace. Přibližuje třídy, které tvoří funkcionalitu a způsob navázání na grafické uživatelské rozhraní. Na konci kapitoly jsou popsány použité technologie v aplikaci. V poslední kapitole 5 je popsáno testování aplikace, odhalené nedostatky v prototypu a návrhy na vylepšení, které vyplynuly s uživatelských poznatků a rad.

1.1 Motivace – Rozšířená realita

V knize *Augmented Reality: An Emerging Technologies Guide to AR* [4] definují rozšířenou realitu poměrně přesně. Uvádí, že:

„Augmented Reality is taking digital or computer generated information, whether it be images, audio, video, and touch or haptic sensations and overlaying them over in a real-time environment.“

Volně přeloženo lze za rozšířenou realitu považovat kombinaci reálného prostředí, do kterého v reálném čase zobrazujeme další prvky pomocí různých prostředků. Například projektoru, brýlí i obyčejného displeje například mobilního telefonu. Můžou se plést pojmy rozšířená realita a virtuální realita. Zásadní rozdíl je v tom, že virtuální realita uzavírá uživatele do počítačem generovaného světa, kdežto rozšířená realita doplňuje skutečný svět o virtuální prvky.



Obrázek 1.1: Rozšířená realita

Kritéria rozšířené reality

Realitu lze považovat za rozšířenou, pokud splňuje následující podmínky (podle [4]):

- Kombinuje skutečné a virtuální skutečnosti/informace
- Reaguje na podněty v reálném čase
- Pracuje v 3D prostoru

Z těchto důvodů nelze považovat za rozšířenou realitu například filmy. Typickým příkladem rozšířené reality pak může být například informační grafika v televizním (sportovním) vysílání. Pro ilustraci může také posloužit obrázek 1.1, který ukazuje využití rozšířené reality v mobilní aplikaci. Mobilní platformy nabízí další možnosti využití rozšířené reality. Například aplikace Face Swap pro Android.

1.2 Existující řešení

Pro srovnání bych rád uvedl dvě řešení stejného problému. Obě řešení volí trochu odlišný způsob zobrazování rozšířené reality, stejně tak i komunikaci mezi hudebním nástrojem a počítačem.

TAP – The Augmented Piano

První ze dvou řešení se jmenuje TAP. Projekt vznikl jako bakalářská práce na Hasselt University a byla vytvořena Linsey Raymaekers pod vedením Jo Vermeulen a prof. dr. Kris Luyten. V rámci práce vzniklo také prezentační video.¹

System umožňuje práci s různým počtem kláves na nástroji, pro komunikaci mezi počítačem a nástrojem je použité MIDI spojení. Cílem práce nebylo nahradit stávající výukové metody ale rozšířit možnosti výuky hry na klavír. Informace byly čerpány z webové stránky projektu [11], která vznikla jako webová prezentace práce [5].

¹<https://www.youtube.com/watch?v=Abstj0PdkW0>

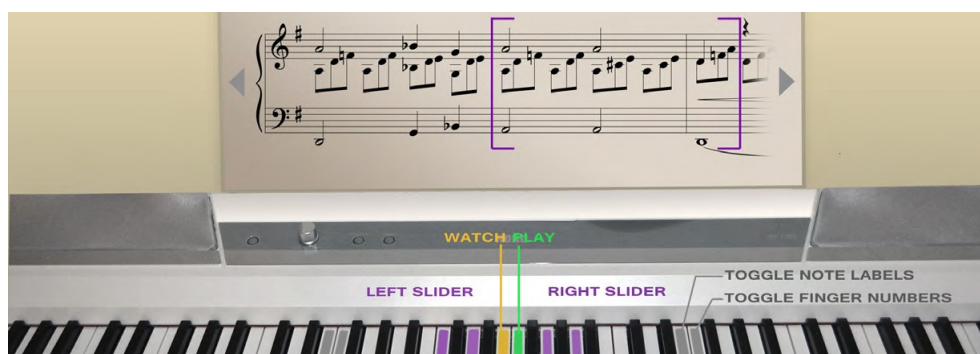


Obrázek 1.2: TAP – z oficiálních stránek

Encore – AR Piano Tutor

Druhé řešení je dílem George Merreta. Oproti prvnímu řešení se liší tím, že pro zobrazování prvků rozšířené reality používá brýle. Pro komunikaci nástroje s počítačem je použita analýza zvukových signálů [7]. Je to sice náročnější řešení na implementaci, ale následně je umožněno použití i na nástrojích, které nemají MIDI rozhraní. Funkčnost je demonstrována pomocí videa².

Dalším rozdílem je ve stylu výuky. Zatímco TAP přehrává celou skladku a reprezentuje ji pomocí barevných bloků nad jednotlivými klávesami, Encore skladbu rozděluje na části, které učí, čímž je bližší běžné výuce.



Obrázek 1.3: Encore AR Piano tutor

²<https://vimeo.com/98980812>

1.3 Porovnání hardware

Tato podkapitola popisuje další varianty řešení hardwaru systému. Ukazuje rozdíly mezi jednotlivými způsoby a přibližuje jejich pozitiva a negativa. Hlavně se věnuje variantám při zobrazování virtuální reality a zpětné vazby od nástroje.

1.3.1 Zobrazení rozšířené reality

Pro zobrazení rozšířené reality lze vzít v úvahu dvě hlavní možnosti. Jednu tvoří klasický dataprojektor, který se hodí k promítání na bílé klapky klavíru. Projektor není těžké sehnat, jedná se o v dnešní době běžnou elektroniku. Nevýhoda pak je dostat projektor do polohy, kdy by promítal kolmo dolů. Běžné uchycení umožňuje projekční rovinu sklonit jen kolem 305° , což nestačí.

Mimo projektor lze použít brýle pro rozšířenou realitu, které se začínají objevovat. Jsou obtížnější k sehnání než projektor a také více stojí. Také mohou působit problémy pro uživatele s běžnými dioptrickými brýlemi, ale tento nedostatek by měl řešit výrobce brýlí. Jako příklad brýlí lze uvést produkt Microsoft Hololens³.

Klady i zápory shrnuje tabulka 1.1.

<i>Projektor</i>	<i>Brýle</i>
+ Snadno dostupný, nižší cena	+ Zabírají méně místa
– Způsob uchycení, vyšší prostorové nároky	– Vyšší cena, horší dostupnost, mohou vadit uživateli

Tabulka 1.1: Srovnání brýlí a projektoru

1.3.2 Příjem zvuku

<i>MIDI</i>	<i>Mikrofon</i>
+ Přesný, rychlejší	+ Funguje s libovolným klávesovým nástrojem
– Lze použít jen pro digitální piano	– Méně přesný, náročnější na implementaci

Tabulka 1.2: Srovnání MIDI a mikrofonu

Pro možnost kontroly hry, případně ovládání aplikace pomocí kláves klavíru, je potřeba rozpoznat, jaká klávesa byla stisknuta. Toho lze docílit dvěma způsoby.

³<https://www.microsoft.com/microsoft-hololens/en-us>

Analýza zvuku z mikrofonu

Je možné pomocí Fourierovy transformace analyzovat zvukový signál z mikrofonu. Na základě této analýzy lze poznat, jaké klávesy byly zmáčknuty. Je ale vyžadován poměrně kvalitní mikrofón, s dobrým stíněním pro přesnější výsledky. Další nepřesnosti pak mohou vzniknout, když bude potřeba rozeznat akordy.

MIDI rozhraní

Na rozdíl od mikrofonu, MIDI může nabídnout přesnější odezvu na stisk kláves. Existují knihovny, které práci s MIDI rozhraním usnadňují, například podpora ve WinApi⁴ V tabulce 1.2 je srovnání mezi oběma zmíněnými způsoby. Pro prototyp v této práci je zvoleno MIDI rozhraní, protože dostupný nástroj ho má k dispozici.

⁴<https://msdn.microsoft.com>

Kapitola 2

Teorie

Druhá kapitola práce přibližuje teoretické poznatky, ze kterých pak vychází celá práce. Nejprve je přiblížena geometrie z pohledu počítačového vidění a je přiblížen termín homografie. Kromě práce s obrazem je přiblížen i standard MIDI pro digitální reprezentaci hudby.

2.1 Projektivní geometrie

Geometrie je matematický vědní obor, který se zabývá tvarem a velikostí obrazců a také vztahy mezi nimi. Základní je Euklidova geometrie, která umožňuje popsat prostor pomocí úseček, přímek a úhlů, které tyto přímky svírají. Platí, že přímky, které jsou rovnoběžné, se nikdy neprotnou.

Problém nastane, když chceme zachytit 3D prostor pomocí kamery (převést na 2D). Na to Euklidova geometrie nestačí, protože se ztrácí úhly mezi přímkami. Rovnoběžné přímky se pak nezobrazí rovnoběžně, ale budou se sbíhat. Je proto potřeba použít projekтивní geometrii, která funguje v n -dimenzionálním prostoru. Euklidova geometrie pak tvoří podmnožinu projekтивní geometrie.

Projektivní geometrie je matematický prostředek, který se využívá hlavně v oblasti počítačového vidění. Největší využití pak nalezneme v oblastech jako kalibrace kamery, rozpoznání objektů v obraze, rekonstrukce scény z 2D obrázků a další. Tato sekce vychází z informací uvedených na [1].

2.1.1 Transformace

Transformace u projekтивní geometrie znamená převedení bodu z projekтивního prostoru do jiného projekтивního prostoru. Pokud chceme transformovat celý objekt, znamená to transformovat každý bod daného objektu.

Různé geometrie mají odlišné možnosti při transformaci bodů, popřípadě objektů. Například obyčejná Euklidova geometrie umožňuje objekty pouze rotovat a posunout. Neumí objekty zvětšovat ani zkosit. To umožňují až afinní transformace.

Projektivní transformace pak umožňuje totéž, co afinní transformace s tím, že umí pracovat s více dimenzemi. Při projekтивní transformaci je zaručeno zachování typu objektu, tedy když transformujeme úsečku, získáme opět úsečku a výskyt¹. To znamená že bod, který se nacházel na úsečce se nepřesune mimo ni.

¹z angl. incidence

Transformace lze vyjádřit pomocí matic různých velikostí, podle dimenze prostoru. Také je nutné bod vyjádřit pomocí homogenních souřadnic. Reprezentace transformací pomocí matic přináší výhodu v podobě jednoduchého kombinování transformací pomocí běžného násobení matic.

2.1.2 Homogenní souřadnice

Homogenní souřadnice je tvar souřadnic, který umožňuje výhodnou reprezentaci bodu pro aplikaci transformace pomocí násobení matic. Bod s kartézskými souřadnicemi $[x, y]$ reprezentujeme jako $[X, Y, \omega]$. ω se nazývá váha bodu a platí, že $X = x/\omega$ a $Y = y/\omega$. Váha bodu ω se většinou volí jako 1, pak výsledné souřadnice můžeme napsat jako $[x, y, 1]$. [12]

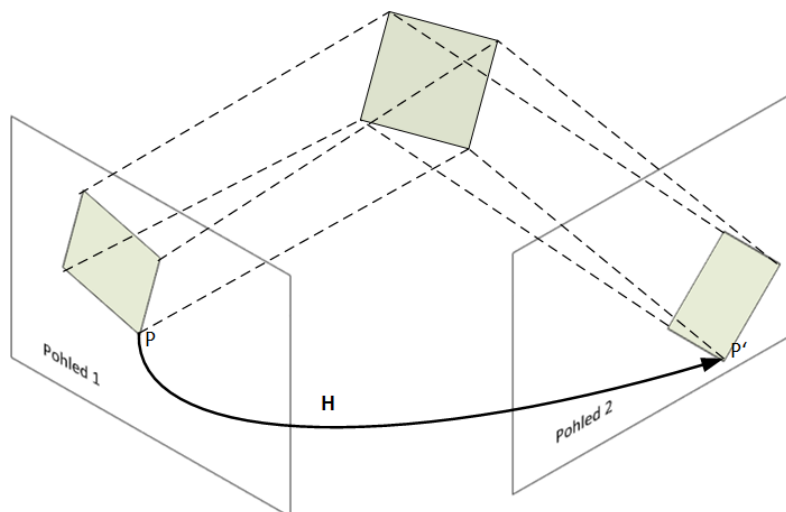
2.2 Homografie

Homografie je matematický termín, který značí mapování bodů jednoho povrchu na body jiného povrchu. V oblasti počítačové grafiky pak mluvíme hlavně o mapování bodů mezi dvěma objekty, kde jeden představuje předmět ve skutečném světě a druhý pak jeho zobrazení v projekční rovině. Toto mapování lze reprezentovat pomocí jedné matice o velikosti 3×3 . Také lze najít homografii pro transformaci objektu z jedné projekční roviny do druhé, jak lze vidět na obrázku 2.1.

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1 \end{bmatrix} \quad (2.1)$$

Tato definice pochází z knihy Learning OpenCV[2].

2.2.1 Mapování pomocí homografie



Obrázek 2.1: Mapování homografií – není potřeba, aby body byly v rozích objektu, ale pro ilustraci byly zvoleny.

Pro nalezení homografie potřebujeme zjistit osm koeficientů matice homografie. Tyto koeficienty vyjadřují vztah mezi objektem v jednom pohledu a dalším objektem v druhém pohledu. Potřebujeme nalézt minimálně osm bodů, čtyři v každém pohledu, které si vzájemně odpovídají. Na základě nich lze koeficienty dopočítat.

Nalezení matice homografie

Informace v této sekci byly čerpány z článku [6]. Pro získání matice homografie 2.1 mějme dvě roviny $R_1(x, y)$ a $R_2(x, y)$. Jednu rovinu pak lze převést pomocí vztahu:

$$R_1(x, y) = R_2(H(x, y)) \quad (2.2)$$

Aby bylo možné transformovat jednu rovinu na druhou, je potřeba mít body roviny zadané pomocí homogenních souřadnic. Matici transformace H pak lze získat postupným skládáním jednodušších matic transformace.

Roviny se od sebe mohou lišit například rotací, kterou reprezentuje matice:

$$M_{rotace} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

dále pak třeba mohou být roviny různě posunuty, posun reprezentuje matice:

$$M_{posun} = \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Kromě předchozích transformací je třeba počítat i s rozdílem velikostí, zkosením a jinou perspektivou. Po složení všech základních transformací do jedné, získáme matici:

$$H = \begin{bmatrix} sa \cos \alpha & -sb \sin \alpha & P_x \\ sa \sin \alpha & sb \cos \alpha & P_y \\ p_0 & p_1 & 1 \end{bmatrix} \quad (2.5)$$

V této matici koeficienty a a b zajišťují zkosení a koeficient s změnu velikosti. Koeficienty p_0 a p_1 popisují rozdíl v perspektivě. Tyto koeficienty lze ve většině případů zanedbat a tím celý další výpočet zjednodušit.

Rovnice pro zjištění homogenních souřadnic bodu $P' \in R_2$ vyhážíme z rovnic:

$$\begin{aligned} x' &= H_{11}x + H_{12}y + H_{13} \\ y' &= H_{21}x + H_{22}y + H_{23} \\ z' &= H_{31}x + H_{32}y + H_{33} \end{aligned} \quad (2.6)$$

DLT algoritmus

Jedná se o algoritmus využívaný v oblasti projektivní geometrie. Často se používá pro určení koeficientů matice homografie.

2.3 Počítačové vidění

Počítačové vidění[2] je část výpočetní techniky, která se zabývá zpracováním obrazu a rozpoznávání obrázků v něm. Největší význam tohle odvětví získává v medicíně, kdy se používá k analýze například rentgenových snímků. Také se může používat pro navigaci autonomních vozidel, rozpoznávání osob na fotkách a jiné.

Počítačové vidění umožňuje přeměnu dat z obrázku nebo videozáznamu na jinou znalost, kdy je program na obrázku schopen například rozeznat a vyznačit obličej nebo na základě informací z obrazu provést příslušnou reakci. Třeba systém řízení autonomního vozidla, který při rozeznání překážky je schopen příslušně reagovat.

Ačkoliv pro člověka je vidění naprosto přirozená dovednost a detekovat v obrazu nějakou informaci nám trvá chvíli bez sebemenší námahy, u počítače je problém. Obraz vidí pouze jako soubor čísel, kde není úplně snadné cokoliv identifikovat.

2.4 MIDI

Zkratka MIDI pochází z anglického „Musical Instrument Digital Interface“ a představuje průmyslový standard pro komunikaci mezi digitálními nástroji a ostatními zařízeními. Ne-specifikuje pouze komunikaci a způsob přenosu dat, ale také způsoby uložení skladeb. Podporuje také celou řadu nástrojů, kterou lze jednoznačně identifikovat. [8]

2.4.1 Komunikace s PC

Pro komunikaci s PC je nutné použít speciální zvukovou kartu, která umí zpracovávat MIDI signál, nebo lze využít připojení přes USB. Pro napojení nástroje na USB je ovšem potřeba převodník, který může být u moderních hudebních nástrojů vestavěn přímo, nebo lze použít standardní MIDI rozhraní 2.2 a signál převádět pomocí speciálního převodníku.

Kromě kabelového přenosu existuje i standard pro přenos přes Bluetooth Low Energy, kterou převážně využívají mobilní zařízení.



Obrázek 2.2: MIDI rozhraní a převodník

Kapitola 3

Návrh řešení

Tato kapitola popisuje návrh prototypu aplikace, která slouží k výuce hry na klavír a využívá k tomu prvky rozšířené reality. První sekce 3.1 představuje hardware a způsob sestavení prototypu. Sekce 3.2 popisuje způsob detekce a zvýraznění kláves nástroje a sekce 3.3 přibližuje způsob komunikace mezi nástrojem a aplikací.

3.1 Popis systému



Obrázek 3.1: Sestavený systém

Cílem práce je vytvořit aplikaci, která by demonstrovala využití rozšířené reality při výuce hry na klavír. Aplikace by měla umožnit přehrát skladbu, kterou si uživatel vybere pomocí menu. Přehrávání by mělo mít nastavitelnou rychlost a také by mělo být umožněno přehrát jen úsek skladby.

Pod přehráním je myšleno zvýraznění kláves na hudebním nástroji. K tomu je potřeba vyřešit barevné odlišení kláves a komunikaci mezi nástrojem a počítačem. V rámci de-

monstrace byly kromě přehrávání přidány další módy: jeden, který označí každou klávesu odpovídajícím písmenem a další, který v reálném čase zvýrazňuje stisknuté klávesy. Ukázka sestaveného systému je na obrázku 3.1.

Klavír

Celý systém se stavěl kolem digitálního pianu značky YAMAHA, model YDP-141C, který poskytuje veškerou potřebnou konektivitu. Nabízí dva výstupy na sluchátka pomocí 6,3mm audio jacku a standardní MIDI rozhraní. Má 88 kláves.

Kamera

Pro zjištění polohy klaviatury byla zakoupena webkamera A4tech PK-910H. Tato kamera umožňuje nahrávání full HD rozlišení. Nejdůležitější vlastnost, pro kterou byla vybrána, je autofokus.

Kamera se v systému nachází dva metry nad zemí a případné ruční ostření obrazu by mohlo dělat problémy. Při takové výšce byl problém i délka kabelu, proto bylo nutné sehnat USB prodlužovací kabel.

Projektor

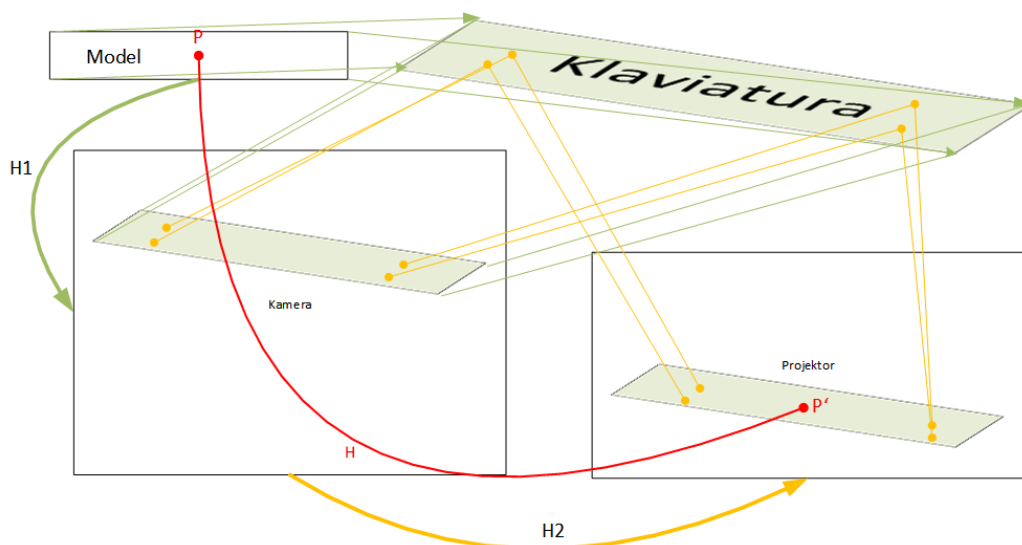
Pro účely práce posloužil filmový projektor Epson EH-TW5210, který poskytuje více než dostačující full HD rozlišení.

Potíže nastaly při snaze zajistit, aby byl projektor otočen směrem dolů a zároveň byl schopen promítat na celou klaviaturu. K tomu byl zapůjčen osvětlovací stojan s maximální výškou 4 metry a vyroben speciální držák, který umožnil ke stojanu přišroubovat projektor.

3.2 Zvýraznění kláves

Aplikace musí zajistit správnou projekci zvýraznění na klaviaturu. Řešení je naznačeno na obrázku 3.2. Nejjednodušším řešením je s využitím homografie, kdy klávesy nástroje označíme za rovinu. Drobný výškový rozdíl mezi bílými a černými klávesami lze zanedbat při minimálních nepřesnostech v konečném mapování.

Princip zvýrazňování funguje tak, že se vytvoří model kláves, který odpovídá počtu a rozložení kláves nástroje, na který chceme kreslit. Program potřebuje vědět, kde přesně se nástroj nachází. K tomu využívá kameru, kterou můžeme reprezentovat jako matici bodů. V tomto obraze lze najít rohové body klaviatury, a pomocí homografie H_1 na ni namapovat model. Dále je potřeba propojit model, respektive obraz z kamery s projektorem. Toho můžeme docílit opět pomocí homografie, tentokrát H_2 kdy projektor na klaviaturu promítne čtyři body a opět jsou význačné body vybrané v pohledu kamery. Protože homografii můžeme reprezentovat maticí, lze jednoduše získat vztah pro mapování modelu přímo na projekční plátno pomocí násobení matic. $H = H_1 * H_2$. Například mapování bodu P z modelu kláves do pohledu projektoru na bod P' lze provést jednoduše pomocí rovnice $P' = H * P$. Celé mapování je v obrázku 3.2 znázorněno červeně.



Obrázek 3.2: Způsob zvýraznění kláves pomocí homografie

3.3 Zpětná vazba od nástroje

Aplikace je výuková, proto by měla být schopna rozpoznat, jaká klávesa byla zahrána. To z důvodu, aby bylo možné hráče upozornit na chybu. Další použití by mohlo být pro ovládání aplikace přímo od nástroje, aby uživatel nemusel obsluhovat aplikaci na počítači. Protože klavír v prototypu disponuje MIDI rozhraním, lze využít právě toto rozhraní a není nutno zabývat se analýzou zvuku, případně detekce stisknuté klávesy přes kameru. Dalším důvodem, proč byla zvolena komunikace přes MIDI bylo, že pro zachycení zvuku nástroje by bylo potřeba počítat s kvalitním mikrofonom, který by zvýšil cenu celého systému.

Kapitola 4

Implementace

Tato kapitola popisuje, jak je implementována demo aplikace pro výukový systém hry na klavír pomocí rozšířené reality. V sekci 4.1 je popsána třída, která reprezentuje model kláves odpovídající návrhu na obrázku 3.2. Sekce 4.2 popisuje, jakým způsobem je řešena komunikace mezi nástrojem a aplikací.

V podkapitole 4.5 je pak popsáno, jakým způsobem je implementováno grafické uživatelské rozhraní. Poslední podkapitola 4.6 jsou popsány použité externí knihovny a nástroje.

4.1 Model kláves

Model kláves reprezentuje třída `KeyboardModel`. V této třídě jsou zapouzdřeny metody a atributy pro vytvoření základního modelu klaviatury.

Při vytváření objektu třídy je použit konstruktor se dvěma parametry, který očekává název klávesy nejvíce vlevo a počet bílých kláves. Tedy volání `KeyboardModel('a', 52)` vytvoří model kláves na obrázku 4.1. V tomtéž obrázku je pak vidět i tvorba modelu pro menší klaviaturu. Vnitřní reprezentace modelu je vytvořena pomocí knihovny `OpenCV` a její třídy `cv::Mat`.

Třída `KeyboardModel` také implementuje metody, které v modelu kláves umožní zvýraznit klávesy. Konkrétně metoda `highlightKeys(std::vector<int>)`. Metoda očekává jeden parametr a to vektor celých čísel, které se mají na modelu zvýraznit. Čísla odpovídají standardním¹ MIDI kódům tónů. Model počítá pouze s oktávami -1 a vyššími. Návrátový typ metody je nový objekt `cv::Mat`, který je stejně velký jako model a obsahuje zvýrazněné klávesy. Na obrázku 4.2 je vidět příklad. Jsou zvýrazněny první dvě klávesy C na malém modelu z obrázku 4.1.

K nalezení homografie a mapování modelu do obrazu projektoru jsou využity funkce z knihovny `OpenCV` a to `cv::findHomography()`. Tato funkce přijímá jako parametry dvě struktury typu `std::vector<cv::Point2f>`. Tyto vektory obsahují dvojice odpovídajících bodů ze dvou pohledů. Funkce umožňuje nastavit způsob výpočtu matice homografie, ale pro prototyp stačilo ponechat ostatní parametry na výchozích hodnotách.

Pro transformaci modelu kláves do pohledu projektoru je použita funkce `cv::warpPerspective()`, která očekává vstupní a výstupní obrázek, matici projektivní transformace a velikost výsledného obrazu. Pro dosažení hezčího efektu zvýraznění bylo potřeba nastavit ve funkci příznak `cv::INTER_AREA`, který zajistí, že funkce při výpočtu nového bodu bere v potaz i okolí zdrojového bodu.

¹<http://computermusicresource.com/midikeys.html>



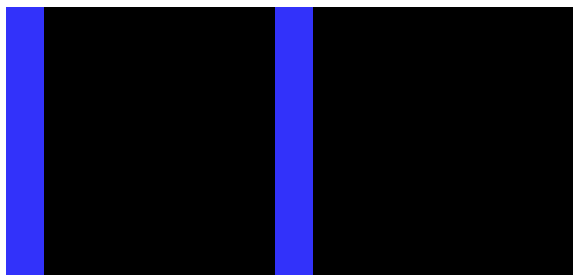
Obrázek 4.1: Model kláves

4.2 Zpětná vazba

Komunikaci přes MIDI rozhraní zajišťuje třída `MidiController`. Třída k tomu využívá `WinAPI`, kvůli které aplikace ztrácí přenositelnost na jiné platformy, než `Windows`. Pro vytvoření objektu třídy se používá explicitní konstruktor, který vytvoří objekt třídy a otevře komunikaci s MIDI zařízením. Před otevřením komunikace je kontrolován počet připojených MIDI zařízení pomocí funkce `midiInGetNumDevs()` z knihovny `WinAPI`. Současná implementace počítá s pouze jedním MIDI zařízením současně. Toto jedno zařízení je otevřeno pomocí funkce `midiInOpen()` opět z knihovny `WinAPI`. Vytvořený objekt se uloží do třídní proměnné `singleton`, aby bylo možné využít systém signálů a slotů v `Qt Frameworku`.

Třída `MidiController` obsahuje hlavní statickou metodu. Metoda `MidiInProc`² je `CALLBACK` funkce, která je volaná při zpracování MIDI zpráv ze zařízení. Metoda přijímá 5 parametrů. Pro implementaci modelu je důležitá „handle“ midi zařízení, kterou získáme při otevření MIDI komunikace. Dále parametr `wMsg`, který určuje typ přijaté zprávy. Jestliže typ zprávy je `MIM_DATA`, tak parametry `dwParam1` obsahuje informaci která klapka byla stisknuta nebo uvolněna. V parametru `dwParam2` získáme čas, který uplynul od začátku naslouchání, tedy od vytvoření spojení. Informace o klávesách se využívá k aktualizování

²<https://msdn.microsoft.com>



Obrázek 4.2: Zvýraznění kláves

vektoru `pressedKeys`. Do vektoru se ukládají aktuálně stisknuté klávesy. Přejde-li zpráva o změně klávesy, která se ve vektoru už nachází, je tato zpráva brána jako informace o uvolnění a daná klávesa je odebrána z vektoru. Do dalšího statického atributu `time` je pak uložen čas poslední změny vektoru.

4.3 Vnitřní reprezentace skladeb

Třída `Song` představuje vnitřní reprezentaci skladby v aplikaci. Skladba je uložena v datové struktuře `QMap` z Qt Frameworku. Výhoda této struktury je, že uchovává pořadí prvků od nejstaršího po nejnovější, což usnadňuje nahrávání a přehrávání.

V třídě jsou implementovány metody `setMIDIIn()` pro nastavení vstupního MIDI zařízení. Tato metoda očekává jako parametr objekt třídy `MidiController`. Pro začátek a konec nahrávání používá systém signálů a slotů z Qt Frameworku.

Data pro objekt třídy `Song` se získávají nahráváním skladby z MIDI rozhraní, které v aplikaci reprezentuje objekt třídy `MidiController`. Po příchodu signálu na slot `startRecord()` se vynuluje časovač MIDI rozhraní a spojí se signál `keysChanged()` třídy `MidiController` se slotem `addTone()`. Metoda obsluhující tento slot zajistí, že do struktury `QMap` data se uloží aktuálně stisknuté klávesy pod časem v milisekundách, který uběhl od začátku nahrávání.

Skladby lze ukládat pomocí metody `saveToFile()` a číst se souboru pomocí metody `loadFromFile()`. Metody vyžadují jako parametr cestu k souboru. Aby bylo možné uložit skladbu do souboru, je nutné ji serializovat. Serializace je zajištěna pomocí operátorů `<<` a `>>`.

4.4 Pomocné třídy

V této sekci jsou popsány ostatní modelové třídy, které nejsou tak rozsáhlé na samostatnou sekci.

Třída `ProjectorModel`

Tato třída se používá pro matice homografie H_2 z obrazu kamery do obrazu projektoru, a matice homografie H , která slouží k mapování modelu přímo do obrazu projektoru. Jediné metody, které tato třída obsahuje jsou „getter“ a „setter“ obou matic.

Třída Tutor

Tato třída slouží k samotnému přehrání skladby. Implementuje metody pro nastavení přehrávané skladby třídy `Song` a okna projektoru, které reprezentuje dále popsaná třída `ProjectorWindow`. Pomocí metody `setSpeed()` je nastavena rychlost přehrávání ve formě desetinného čísla s dvojitou přesností.

Samotné přehrávání se spustí pomocí metody `playSong()`. V této metodě se nastaví vnitřní iterátor na počátek skladby a vyšle se signál na slot `showNextState()`, který zpracuje další položku z proměnné `QMap data` z aktuálně přehrávané skladby. Pokud se jedná o poslední položku v mapě, je vyslán signál `playbackEnd()`, jinak se naplánuje zobrazení dalšího stavu pomocí statické metody `singleShot()` ze třídy `QTimer`. Tato metoda vyžaduje zadat čas, za jak dlouho se má vyslat signál. Tento čas se vypočítává z klíče aktuální a následující položky. Rozdíl obou časů je ještě násoben koeficientem rychlosti pro možnost změny rychlosti přehrávání.

4.5 Třídy uživatelského rozhraní

V této sekci jsou popsány jednotlivé třídy grafického uživatelského rozhraní a práce s ním. Jako první je popsáno hlavní okno aplikace 4.5.1, pak okno pro práci s kamerou 4.5.2 a okno pro práci s projektorem 4.5.3.

Kromě oken se uživatelského rozhraní týká i třída `Converters` 4.4, která obsahuje metodu pro konverzi obrázku z OpenCV na obrázek Qt Frameworku a třída `DragLabel` 4.5.4 s vlastní implementací grafického „widgetu“ pro Qt framework.

4.5.1 Třída MainWindow



Obrázek 4.3: Hlavní okno aplikace

Třída `MainWindow` reprezentuje hlavní okno aplikace. To se skládá z levého panelu s nabídkou a pravé části, kde se zobrazují podnabídky dané části, jak lze vidět na obrázku 4.3

Cílem tohoto okna je propojit všechny části aplikace a umožnit její snadnou obsluhu. K tomuto účelu jsou využity grafické prvky Qt frameworku a také signály a sloty pro komunikaci mezi nimi navzájem a modelovými třídami.

Kalibrace systému

Z hlavního okna aplikace lze zahájit konfiguraci celého systému, kterou je potřeba udělat manuálně. Pomocí tlačítka **Nová konfigurace**, které vyšle signál na slot `newConfigCreate()`, čímž se vytvoří okna pro práci s kamerou třídy `CameraWindow` a okno projektoru `ProjectorWindow`. Samotná konfigurace pak probíhá pomocí nově otevřených oken. Nejprve je v okně kamery nutno vyznačit rohy klaviatury, počínaje levým horním rohem z pohledu hráče. Další roh je pravý horní, pak levý spodní a pravý spodní. Po označení všech čtyř bodů lze pomocí tlačítka **Najít homografii 1** zjistit vztah pro mapování modelu do obrazu kamery. Jako utvrzení správně nalezeného vztahu se zobrazí okno s modelem mapovaným v pohledu kamery.

Pro nalezení vztahu $H2$ je nutné přemístit okno projektoru 4.5 na plochu, kterou promítá projektor. Okno projektoru lze přepnout do celoobrazovkového režimu. V okně projektoru se nachází červený rámeček, který je potřeba umístit tak, aby obsahoval celou plochu klaviatury. V okně kamery je pak potřeba označit vysvícené body v červeném rámečku ve stejném pořadí jako v případě označení klaviatury. Tedy levý horní, pravý horní, levý spodní a pravý spodní. Volbu lze potvrdit tlačítkem **Najít homografii 2**. Po správné konfiguraci se na klaviatuře nástroje zobrazí jména kláves pro každou bílou klapku.

Práce se skladbami

Na záložce **Tutor** v levé nabídce se nachází rozhraní pro práci se skladbami. Nachází se zde tlačítka pro načtení skladby se souboru a také pro její uložení. Jsou k tomu využívány standardní systémové dialogy pro práci se soubory. Pod těmito tlačítky se nachází tlačítka pro přehrání aktuálně načtené skladby, tlačítko pro zastavení nahrávání a tlačítko pro jeho spuštění. Zbytek panelu zabírá posuvník pro nastavení rychlosti přehrávání pro usnadnění výuky složitější části skladby.

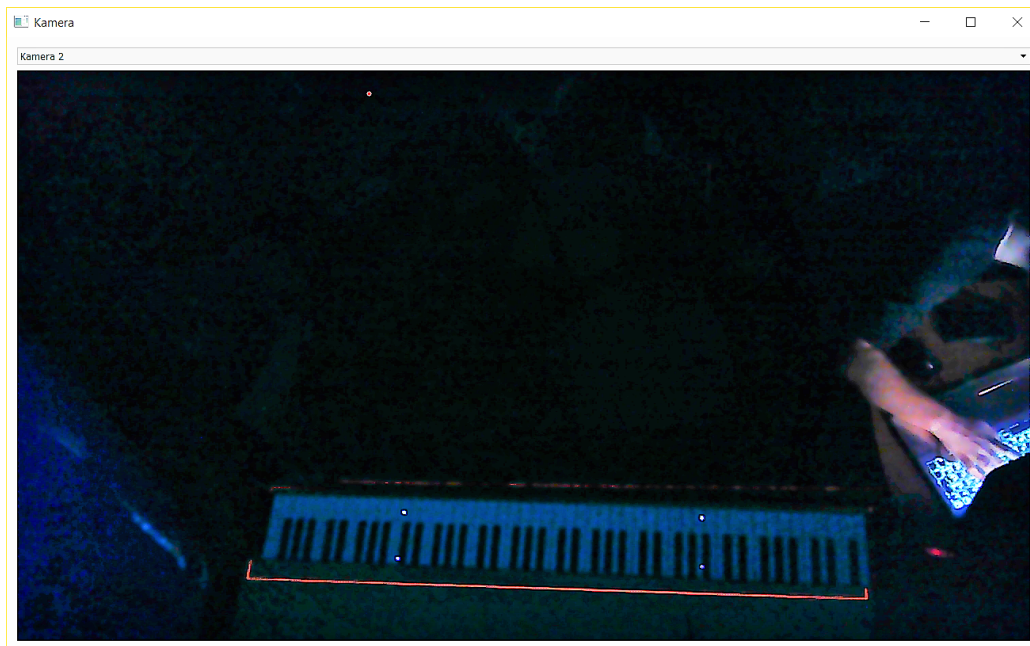
Další funkce

Kromě konfigurace systému spojuje hlavní okno i všechny varianty práce s aplikací. Na záložce **Demo** z levé kontextové nabídky se nachází tlačítka pro začátek a konec nahrávání skladby, přehrání takhle nahrané skladby a dema, které zvýrazní stisknuté klávesy.

4.5.2 Třída `CameraWindow`

Reprezentuje okno pro práci s kamerou. V horní části okna se nachází výběr kamery. Prototyp aplikace umožňuje výběr pouze ze dvou kamer. Většinu okna pak zabírá obraz z kamery, jak je vidět na obrázku 4.4

Obraz z kamery je získán pomocí třídy `cv::VideoCapture` z knihovny OpenCV. Demo aplikace očekává kameru, která je schopna pořídít HD záznam. Zobrazovací frekvence obrazu s kamery je nastavena na 30 snímků za sekundu. Snímky z kamery je před zobrazením



Obrázek 4.4: Okno pro práci s kamerou

nutno překonvertovat na `QImage`, který lze zobrazit přes `QLabel` prvek. K tomu slouží metoda `convertOpenCVMatToQtQImage` z třídy `Converters`.

Je potřeba pracovat s obrazem kamery pomocí myši, pro konfiguraci je potřeba vyznačit v obraze význačné body. K tomu je použita třída `MyEventFilter`, která je definována ve stejném hlavičkovém souboru, jako okno kamery. Tento filtr je aplikovaný na prvek `QLabel`, ve kterém lze vidět obraz kamery a zachytává události způsobené myší. Zvolené body se ukládají do vektoru `selectedPoints`, ale ne více než čtyři. Když je zvolen pátý bod v pořadí, celý vektor se vyčistí a zvolený bod se počítá jako první. Toto opatření bylo zavedeno z důvodu překliknutí a možnosti opravy. Body z vektoru jsou vykreslovány do obrazu kamery pro snadnější orientaci ve výběru.

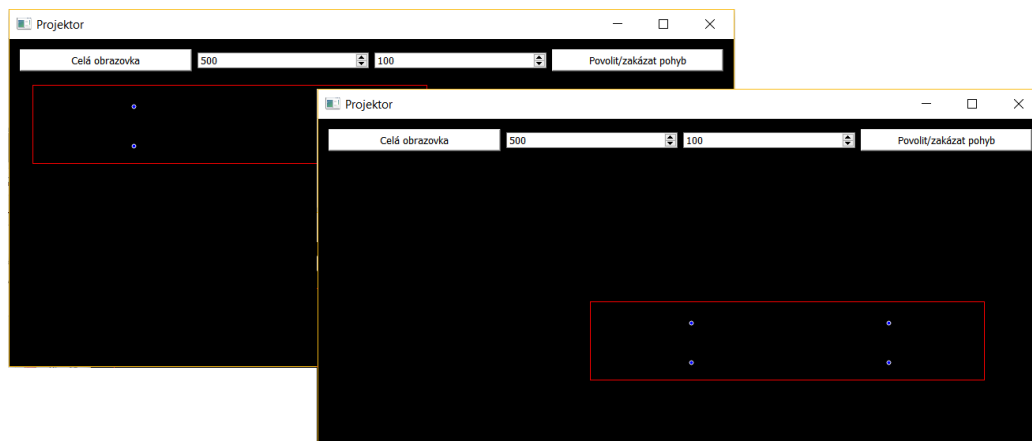
4.5.3 Třída `ProjectorWindow`

Tato třída představuje okno pro kontrolu obrazu projektoru. V horní části je nabídka, která obsahuje tlačítko pro přepnutí okna do režimu celé obrazovky. Dále se zde nachází dva celočíselné vstupy pro nastavení velikosti červeného rámečku ve spodní části okna. Poslední tlačítko v liště povoluje a zakazuje pohyb zmíněného rámečku.

Červený rámeček je znázornění třídy `DragLabel` a ohraničuje místo, kde se očekává klaviatura nástroje, na kterou se bude promítat.

4.5.4 Třída `DragLabel`

Třída `DragLabel` dědí od třídy `QLabel` a rozšiřuje jeho funkcionalitu o možnost libovolného pohybu v rámci nadřazeného objektu třídy `QWidget`. Pohyb objektu třídy `DragLabel` lze ovlivnit pomocí slotu `toggleLock()`, který buď zamkne nebo odemkne možnost pohybování s objektem. Pohyb je zajištěn pomocí metod `mousePressEvent()`, která získá pozici kliknutí a `mouseMoveEvent()`, která posune objektem. Obě třídy přepisují metody se stejným



Obrázek 4.5: Okno pro práci s projektorem

názvem z nadřazené třídy. Metoda `drawPoints()` zajišťuje vykreslení význačných bodů pro propojení pohledu kamery a pohledu projektoru.

4.6 Technologie

Tato podkapitola popisuje některé z technologií, pomocí kterých byla vytvořena demo aplikace.

OpenCV

OpenCV[9] je knihovna, která poskytuje prostředky pro implementaci aplikací, které pracují s počítačovým viděním. Obsahuje přes 2500 optimalizovaných algoritmů pro detekci pohybu ve videu, detekci objektů v obraze, pro kalibraci kamery a jiné.

Knihovna je implementována v jazyce C++ a šířena s přístupným zdrojovým kódem pod tříbodovou licencí BSD. Kromě jazyka C++ knihovna podboruje také další jazyky, jako Python, Javu, C a Matlab. Existuje také Egmucv³, což je .NET/C# adaptér pro využití knihovny OpenCV v jazyce C. Díky široké podpoře jazyků lze vytvářet aplikace pro velké množství zařízení. OpenCV podporuje vývoj pro Windows, Linux i iOS, z mobilních platforem pak Android.

Qt framework

Qt framework[10] umožňuje tvorbu multi-platformních, grafických i textových, uživatelských rozhraní. Podle údajů na oficiální webové stránce jej používá přes milion uživatelů. Součástí Qt frameworku je také celá řada nástrojů, které usnadňují návrh uživatelských rozhraní, například Qt Designer⁴. Veškeré nástroje jsou pak kombinovány do IDE pod jménem Qt Creator [3].

³http://www.emgu.com/wiki/index.php/Main_Page

⁴<http://doc.qt.io/qt-5/qtdesigner-manual.html>

Kapitola 5

Testování

Tato kapitola shrnuje průběh a výsledky testování prototypu aplikace na reálných uživateli. Podkapitola 5.1 přibližuje způsob testování a reprezentuje jeho výsledky. Další podkapitola 5.3 poukazuje na chyby, které byly známy před testováním i ty, které se objevily až v průběhu. Poslední podkapitola 5.4 obsahuje seznam možných vylepšení, které by systému prospěla.

5.1 Uživatelské testování

Testování aplikace probíhalo formou dialogu se zástupci potenciálně možných uživatelů aplikace. Byli vybráni lidé, kteří hrát na klávesové nástroje umí obstojně i tací, kteří neumí hrát na žádný hudební nástroj, ale přesto projevíli zájem aplikaci vyzkoušet.

Každému uživateli bylo předvedeno, jak se s aplikací pracuje. Byl vysvětlen způsob konfigurace a měřen čas, za jak dlouho je dotyčná osoba systém nastavit. Doba nastavení byla testována jen ze zvědavosti. V praxi by se konfigurace neměla provádět moc často. Bylo by vhodnější implementovat automatickou konfiguraci. Data v tabulce 5.1 ukazují průměrné hodnoty pěti uživatelů z 10 měření. Poslední pokusy byly většinou nejrychlejší, protože už byl postup naučený. Také přesnost konfigurace se postupně zvyšovala.

Hráč	Doba konfigurace
1	2 minuty
2	5 minut
3	3,5 minuty
4	7 minut
5	12 minut

Tabulka 5.1: Čas konfigurace

5.1.1 Herní úkoly

Kromě konfigurace bylo dalším úkolem zahrát stupnici C-DUR. Pro tento účel byl předem vytvořen záznam této stupnice pro systém rozšířené reality. Hráči, kteří na klavír hrát umí, nepotřebovali asistenci systému a stupnici zahráli z paměti, což se vzhledem k triviálnosti úkolu dalo očekávat. Ti, kteří u nástroje nikdy neseděli, si načetli stupnici ze souboru a pomocí prvků rozšířené reality pochopili, co bylo potřeba zahrát.

Při testu bylo zjišťováno jestli lze prototyp použít k reálné výuce. Uživatelé byli rozděleni podle úrovně a podle úrovně se lišil i základní požadavek. Začátečníci, kteří u nástroje seděli prakticky poprvé, měli za úkol stupnici přehrát alespoň tak rychle, jak byla nahrána. Tento požadavek všichni začátečníci splnili do deseti minut. Vysvětlení prstokladu pomohlo začátečníkům k přehrání záznamu běžnou rychlostí.

Úroveň hráče	Nejvyšší rychlost	Doba cvičení
začátečník	1x	do 10 minut
pokročilý	2x	ihned
žák ZUŠ	1,2x	do 5 minut

Tabulka 5.2: Učení stupnice

Dalším úkolem bylo přehrát akordy k stupnici C-DUR. Konkrétně tonický, dominantní a subdominantní kvintakord. Opět byl předem připraven záznam akordů. Problémem začátečníků bylo, že nevěděli, které akordy vlastně hrají.

Úroveň hráče	Nejvyšší rychlost	Doba cvičení
začátečník	1x	do 3 minut
pokročilý	3x	ihned
žák ZUŠ	1,5x	do minuty

Tabulka 5.3: Akordy ke stupnici

Posledním cvičením bylo zkoušení obrátů akordu C. Zpočátku působila potíže koordinace prstů na klaviatuře, pokud hráč neměl předchozí zkušenosti s hraním na klávesové nástroje. Po několikátém opakování, případně zpomalení přehrávání se tento nedostatek odstranil.

Úroveň hráče	Nejvyšší rychlost	Doba cvičení
začátečník	1x	do 15 minut
pokročilý	2x	ihned
žák ZUŠ	1,1x	do 5 minut

Tabulka 5.4: Obraty akordu C

5.2 Uživatelské hodnocení

Součástí testování byl i krátký formulář, který zjišťoval reakce uživatelů na prototyp systému. Byly položeny následující otázky, které uživatel ohodnotil známkou 1-10, kdy 10 znamenalo nejlepší hodnocení.

Kladené otázky

1. Jak složité Vám připadalo ovládání aplikace?
2. Jak velký přínos vidíte v aplikaci?
3. Do jaké míry Vás práce s aplikací bavila?

4. Jak byste ohodnotili vzhled aplikace?

Průměrné odpovědi na otázky jsou shrnuty v následující tabulce.

Číslo otázky	Průměrná hodnota odpovědi
1	7
2	6
3	7
4	4

Tabulka 5.5: Uživatelské hodnocení

5.3 Odhalené nedostatky

Při běžné výuce hry na klavír je velmi důležité, jak hráč u nástroje sedí, jakým způsobem má položené ruce na klaviatuře a jak dodržuje předepsaný prstoklad. Toto všechno není možné systémem sledovat a hráče opravovat. Je nutná asistence pedagoga.

Kromě předchozích nedostatků v kontrole techniky, se u prototypu projevuje problém při odlišení levé a pravé ruky. Protože je vše zvýrazňováno jednou barvou, při složitějších skladbách, než jsou jen stupnice a akordy, dochází k prolínání doprovodu a melodie a je těžké určit, co kterou rukou hrát. To by mohlo být vyřešeno tak, že při zvýraznění klávesy by se i napsalo, který prst by měl danou klávesu zahrát. Odlišení levé a pravé ruky by šlo řešit pomocí současného přehrávání dvou „stop“ současně. Každé by šlo nastavit jiná barva zvýraznění.

Dalším nedostatkem je, že prototyp neukazuje chybně stisknuté klávesy při přehrávání skladby. Například klávesa, která by v daný moment neměla znít, ale zní, by se zvýraznila červeně. V prototypu žádné upozornění na chybu není.

5.4 Další vylepšení

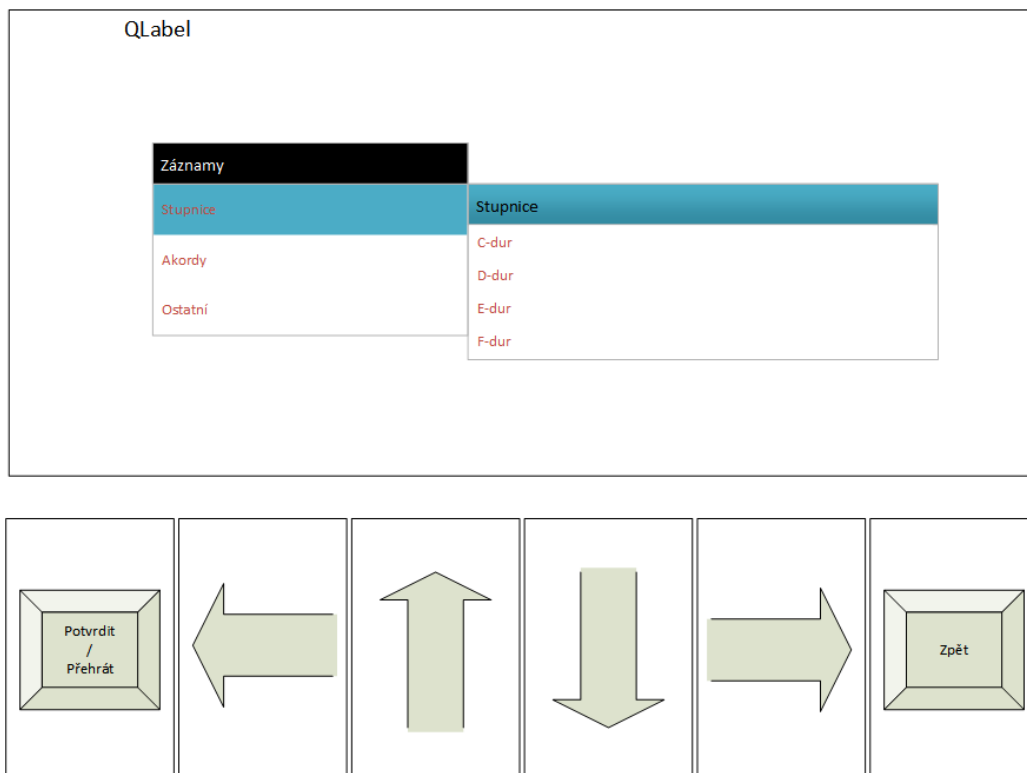
Z uživatelské zpětné vazby vyšlo několik důležitých poznatků, které by celou aplikaci vylepšili.

Ovládání aplikace přes klávesy

Kromě připomínek ke vzhledu aplikace na počítači se objevovali požadavky na ovládání aplikace pomocí klaviatury nástroje. Obrázek 5.1 ukazuje návrh jednoduchého uživatelského rozhraní pro zobrazení na nástroji.

Zobrazování grafických prvků by se dalo řešit pomocí přidání nového prostoru do okna `ProjectorWindow`, kde už se nachází prostor pro promítání na nástroj. Uživatel u nástroje už potřebuje pouze vybrat skladbu, která se má přehrávat, případně nahrát další. Výběr skladeb by šel implementovat pomocí jednoduchého kontextového menu. Toto menu by se ovládalo pomocí kláves, kterým by byla přiřazena speciální funkce jako směrové šipky, potvrzení výběru a zrušení výběru. Funkční klávesy by pak byly odlišeny speciálními popisky.

Obsluhu kláves se speciální funkcí by zajišťoval objekt třídy `MidiController`, který by porovnával informace o stisknutých klávesách s jejich významem. Implementovat by to bylo možné pomocí přepínání dvou módů, kdy v jednom se berou všechny klávesy jako jednotlivé tóny. V druhém módu by měli některé klapky speciální význam.



Obrázek 5.1: Návrh uživatelského rozhraní a ovládacích prvků pro ovládání aplikace pomocí nástroje.

Přehrávání standardních MIDI souborů

Přehrávání MIDI souborů by byl ideální doplněk funkčnosti aplikace. Značně by se usnadnilo získávání skladeb pro výuku. Aplikace pro rozšířenou realitu by pak mohla fungovat podobně, jako program Synthesia, který se používá pro video návody jak zahrát skladbu. Tyto návody se pak šíří pomocí například prostřednictvím serveru YouTube a běžně je využívají hráči, kteří se na klavír učí hrát doma.

Čtení standardních MIDI souborů není nijak složitá záležitost a existují pro to již hotové a funkční knihovny. Jmenovitě například *Midifile*¹ který je speciálně pro čtení MIDI souborů. Ve frameworku *Juce*² se nachází také podpora pro práci s MIDI soubory i rozhraním. Dalším příkladem může být knihovna *RtMidi*³.

¹<http://midifile.sapp.org/>

²<https://www.juce.com/>

³<https://www.music.mcgill.ca/~gary/rtmidi/>

Kapitola 6

Závěr

Cílem práce bylo navrhnout a sestavit systém, který pomocí prvků rozšířené reality podporuje výuku hry na klavír a klávesové nástroje. Pro splnění cíle bylo potřeba nastudovat principy rozšířené reality, seznámit se s homografií a s knihovnou OpenCV. Pro vylepšení zpětné vazby bylo nastudováno použití MIDI rozhraní. Při testování bylo zjištěno, že implementovaný prototyp aplikace lze použít jako doplňující aktivita pro běžnou výuku hry na klavír, ale ovládání v současné podobě není úplně intuitivní.

Při návrhu, který lze rozdělit na návrh systému jako takového a aplikace, bylo nejprve vyřešeno samotné sestavení všech komponentů. Pro implementaci aplikace bylo nutné nastudovat Qt framework a jeho metodiku návrhu, hlavně systém signálů a slotů.

Výsledkem implementace je aplikace, která umožňuje s pomocí projektoru a kamery získat oblast klaviatury nástroje a získat polohu jednotlivých kláves. Tyto klávesy lze jednotlivě zvýraznit. Je implementováno zvýrazňování v reálném čase, kdy klávesy, které jsou hrány, jsou také zvýrazněny. Ovšem pouze za předpokladu přítomnosti MIDI rozhraní. Aplikace umožňuje záznam hry a také přehrání tohoto záznamu různou rychlostí, čímž je umožněna výuka dané skladby. Výsledek je demonstrován na plakátu, který se nachází na přiloženém CD. Stejně tak je na přiloženém CD zdrojový kód aplikace.

Současnou verzi aplikaci lze vylepšit přidáním podpory pro standardní MIDI soubory, čímž by se velmi zvětšila zásoba skladeb k výuce. Také by mohlo být uživateli usnadněna práce s celým systémem. Například umožněním výběru skladeb přímo pomocí klaviatury nástroje. Hodit by se také mohla možnost barevného odlišení jednotlivých kláves při zvýrazňování. Tím by bylo umožněno ukázat, jakým prstokladem nejlépe zahrát přehrávanou skladbu, pokud každému prstu přiřadíme jednu barvu.

Dále by bylo vhodné rozlišit stopu pro levou a pravou ruku. Opět by se mohlo jednat o barevné rozlišení. Oddělení obou ruk by také usnadnilo nácvik hry jedné a poté druhé ruky. Nakonec obou dohromady. Tento způsob výuky je bližší současné běžné výuce.

Oblast rozšířené reality má v době psaní práce velký potenciál a stále se rozvíjí, hlavně v podobě aplikací na mobilní platformy. Jsou představovány nové a nové aplikace a způsoby využití.

Literatura

- [1] Bircheld, S.: *An Introduction to Projective Geometry (for computer vision)*. [Online; navštíveno 27.04.2016].
URL <http://robotics.stanford.edu/~birch/projective/>
- [2] Bradski, G. R.; Kaehler, A.: *Learning OpenCV*. O'Reilly, 2008, ISBN 978-0-596-51613-0.
- [3] Chroboczek, M.: *Grafická uživatelská rozhraní v Qt a C++: [plně kompatibilní s Qt 5]*. Computer Press, Brno, 2013, ISBN 978-80-251-4124-3, první vydání.
- [4] Kipper, G.; Rampolla, J.: *Augmented Reality: An Emerging Technologies Guide to AR*. Elsevier Science, 2012, ISBN 978-1-59749-733-6, 1–27 s., první vydání.
- [5] Raymaekers, L.: *Learning Piano Through an Augmented Piano System*. Universiteit Hasselt, 2013, [Bakalářská práce, <http://www.scriptiebank.be/scriptie/learning-piano-through-augmented-piano-system>].
- [6] van der Walt, S.: *Homography Estimation*. 2010, [Online, navštíveno 10.05.2016].
URL http://dip.sun.ac.za/~stefan/TW793/attach/notes/homography_estimation.pdf
- [7] WWW stránky: *Encore – AR Piano Tutor*. [Online; navštíveno 22.04.2016].
URL <http://www.glm-design.co.uk/?portfolio=encore-ar-piano-tutor>
- [8] WWW stránky: *MIDI association*. [Online; navštíveno 29.04.2016].
URL <https://www.midi.org/specifications>
- [9] WWW stránky: *OpenCV knihovna* . [Online; navštíveno 20.04.2016].
URL <http://opencv.org/about.html>
- [10] WWW stránky: *Qt framework* . [Online; navštíveno 20.04.2016].
URL <https://www.qt.io/company/>
- [11] WWW stránky: *TAP - The Augmented Piano* . [Online; navštíveno 22.04.2016].
URL <http://research.edm.uhasselt.be/~tap/>
- [12] Žára, J.: *Moderní počítačová grafika*. Computer Press, Brno, 2004, ISBN 80-251-0454-0, druhé vydání, strana 542.

Přílohy

Seznam příloh

A Obsah CD

30

Příloha A

Obsah CD

- /demo/ – Složka se skladbami pro testování
- /doc/ – Tato zpráva a její zdrojový tvar
- /poster/ – Plakát a jeho zdrojový tvar
- /src/ – Zdrojový kód aplikace